

A



**AS**

**COMPUTER SCIENCE**

**Paper 2**

**7516/2**

**Insert**

**TABLE 1 for use in answering Questions 9.1  
and 9.2**

**[Turn over]**

**TABLE 1 – standard AQA assembly language instruction set**

LDR Rd, <memory ref>	<b>Load the value stored in the memory location specified by &lt;memory ref&gt; into register d.</b>
STR Rd, <memory ref>	<b>Store the value that is in register d into the memory location specified by &lt;memory ref&gt;.</b>
ADD Rd, Rn, <operand2>	<b>Add the value specified in &lt;operand2&gt; to the value in register n and store the result in register d.</b>
SUB Rd, Rn, <operand2>	<b>Subtract the value specified by &lt;operand2&gt; from the value in register n and store the result in register d.</b>
MOV Rd, <operand2>	<b>Copy the value specified by &lt;operand2&gt; into register d.</b>

CMP Rn, <operand2>	<b>Compare the value stored in register n with the value specified by &lt;operand2&gt;.</b>
B <label>	<b>Always branch to the instruction at position &lt;label&gt; in the program.</b>
B<condition> <label>	<b>Branch to the instruction at position &lt;label&gt; if the last comparison met the criterion specified by &lt;condition&gt;.</b> <b>Possible values for &lt;condition&gt; and their meanings are:</b> EQ: equal to NE: not equal to GT: greater than LT: less than

**TABLE 1 continues on the next page**

**[Turn over]**

AND Rd, Rn, <operand2>	<b>Perform a bitwise logical AND operation between the value in register n and the value specified by &lt;operand2&gt; and store the result in register d.</b>
ORR Rd, Rn, <operand2>	<b>Perform a bitwise logical OR operation between the value in register n and the value specified by &lt;operand2&gt; and store the result in register d.</b>
EOR Rd, Rn, <operand2>	<b>Perform a bitwise logical XOR (exclusive or) operation between the value in register n and the value specified by &lt;operand2&gt; and store the result in register d.</b>

MVN Rd, <operand2>	<b>Perform a bitwise logical NOT operation on the value specified by &lt;operand2&gt; and store the result in register d.</b>
LSL Rd, Rn, <operand2>	<b>Logically shift left the value stored in register n by the number of bits specified by &lt;operand2&gt; and store the result in register d.</b>
LSR Rd, Rn, <operand2>	<b>Logically shift right the value stored in register n by the number of bits specified by &lt;operand2&gt; and store the result in register d.</b>
HALT	<b>Stops the execution of the program.</b>

**[Turn over]**

**BLANK PAGE**

**LABELS:** A label is placed in the code by writing an identifier followed by a colon (:). To refer to a label the identifier of the label is placed after the branch instruction.

### **INTERPRETATION OF <operand2>**

**<operand2> can be interpreted in two different ways, depending on whether the first character is a # or an R:**

- **# – use the decimal value specified after the #, eg #25 means use the decimal value 25**
- **R<sub>m</sub> – use the value stored in register m, eg R6 means use the value stored in register 6**

**The available general purpose registers that the programmer can use are numbered 0–12**

**END OF INSERT**

# BLANK PAGE

## Copyright information

For confidentiality purposes, all acknowledgements of third-party copyright material are published in a separate booklet. This booklet is published after each live examination series and is available for free download from [www.aqa.org.uk](http://www.aqa.org.uk).

Permission to reproduce all copyright material has been applied for. In some cases, efforts to contact copyright-holders may have been unsuccessful and AQA will be happy to rectify any omissions of acknowledgements. If you have any queries please contact the Copyright Team.

Copyright © 2022 AQA and its licensors. All rights reserved.

**IB/M/CD/Jun22/7516/2/E2**



2 2 6 A 7 5 1 6 / 2